

Дано два файла – MainProgramm и лаунчер

Пробуем запустить MainProgramm

```
Enter key: 121234
INVALID
Access denied. Launcher did not authorize (exit code = 1).
```

Видимо, она что-то делает с лаунчером

Пытаемся запустить лаунчер

```
Enter key: 23123|
```

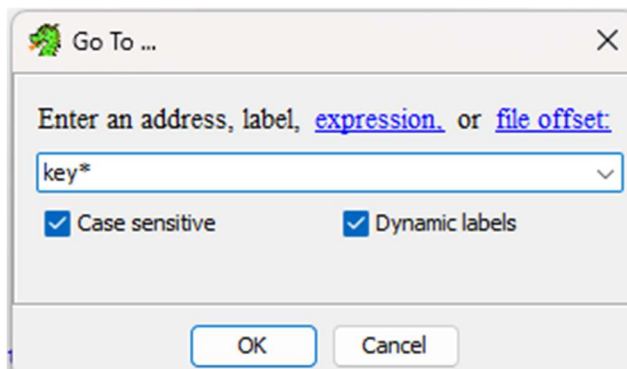
Требуется запустить ключ полностью аналогичным выводом и после ввода закрывается

Идём в гидру смотреть лаунчер

Сразу найденная точка входа ничего не даёт

```
1
2 /* Library Function - Single Match
3     int __cdecl __srt_common_main(void)
4
5     Libraries: Visual Studio 2017 Debug, Visual Studio 2019 Debug */
6
7 int __cdecl __srt_common_main(void)
8
9 {
10     int iVar1;
11
12     __security_init_cookie();
13     iVar1 = __srt_common_main_seh();
14     return iVar1;
15 }
16
```

Вспоминаем вывод в консоль, ищем



Находим дизассемблированный участок, там присутствует XREF[1]: FUN...

1400227c0	45 6e 74	ds	"Enter key: "	XREF[1]:	FUN_140017f10:140017f58(*)
	65 72 20				
	6b 65 79 ...				
1400227cc	00	??	00h		
1400227cd	00	??	00h		
1400227ce	00	??	00h		
1400227cf	00	??	00h		

Идем туда, смотрим дизассемблированный код. Сразу можно заметить ветвление на OK и INVALID-статусы проверки

```

if (cVar2 == '\0') {
    thunk_FUN_140012030((longlong *)cout_exref, "INVALID\n");
    local_64 = 1;
    thunk_FUN_140014f60(local_200);
    thunk_FUN_140014f60(local_240);
}
else {
    thunk_FUN_140012030((longlong *)cout_exref, "OK\n");
    local_84 = 0;
    thunk_FUN_140014f60(local_200);
    thunk_FUN_140014f60(local_240);
}

```

Планируем пропатчить лаунчер, чтобы проверка всегда была успешной

Ищем адрес инструкции

140018017	0f b6 c0	MOVZX	EAX, AL		
14001801a	85 c0	TEST	EAX, EAX		
14001801c	74 3e	JZ	LAB_14001805c		
14001801e	48 8d 15	LEA	RDX, [DAT_1400227d0]	= 4Fh	0
	ab a7 00 00				
140018025	48 8b 0d	MOV	RCK, qword ptr [->MSVCF140D.DLL:std::cout]	= 0002c0b4	
	dc 31 01 00				
14001802c	e8 8d 90	CALL	thunk_FUN_140012030	undefined thunk_FUN_14001	
	ff ff				
140018031	90	NOP			
140018032	c7 85 c4	MOV	dword ptr [RBP + local_84], 0x0		
	01 00 00				

Имеет опкод 74 3e, находится по адресу 1801c, так как 1400 – виртуальная часть, в системе база модуля будет загружена по другому адресу, но в этом модуле нужная инструкция будет находится именно по 1801c

Идём в x64dbg. Открываем лаунчер, жмем Alt+e, ищем модуль launcher.exe имеет базу 00007FF6F6620000. Прибавляем к нему 1801c, получим 00007FF6F663801C

Идем по этому адресу, видим следующее

```
74 3E je launcher.7FF6F663805C jump sho
48:8D15 ABA70000 lea rdx,qword ptr ds:[<"OK\n"...> ] main.cpp
48:8B00 DC310100 mov rcx,qword ptr ds:[<class std::basic_ostream<char, struct std:: rcx:"MZ"
E8 8D90FFFF call launcher.7FF6F66310BE calls a
90 nop no opera
C785 C4010000 0000 mov dword ptr ss:[rbp+1C4],0 main.cpp
48:8D4D 48 lea rcx,qword ptr ss:[rbp+48] load eff
E8 FB90FFFF call launcher.7FF6F6631140 calls a
90 nop no opera
48:8D4D 08 lea rcx,qword ptr ss:[rbp+8] [rbp+08]
E8 F190FFFF call launcher.7FF6F6631140 calls a
8B85 C4010000 mov eax,dword ptr ss:[rbp+1C4] moves da
E9 9B000000 jmp launcher.7FF6F66380F5 jump
EB 39 jmp launcher.7FF6F6638095 main.cpp
48:8D15 75A70000 lea rdx,qword ptr ds:[<"INVALID\n"...> ] main.cpp
```

Чтобы не было перехода, нужно заменить je на jne

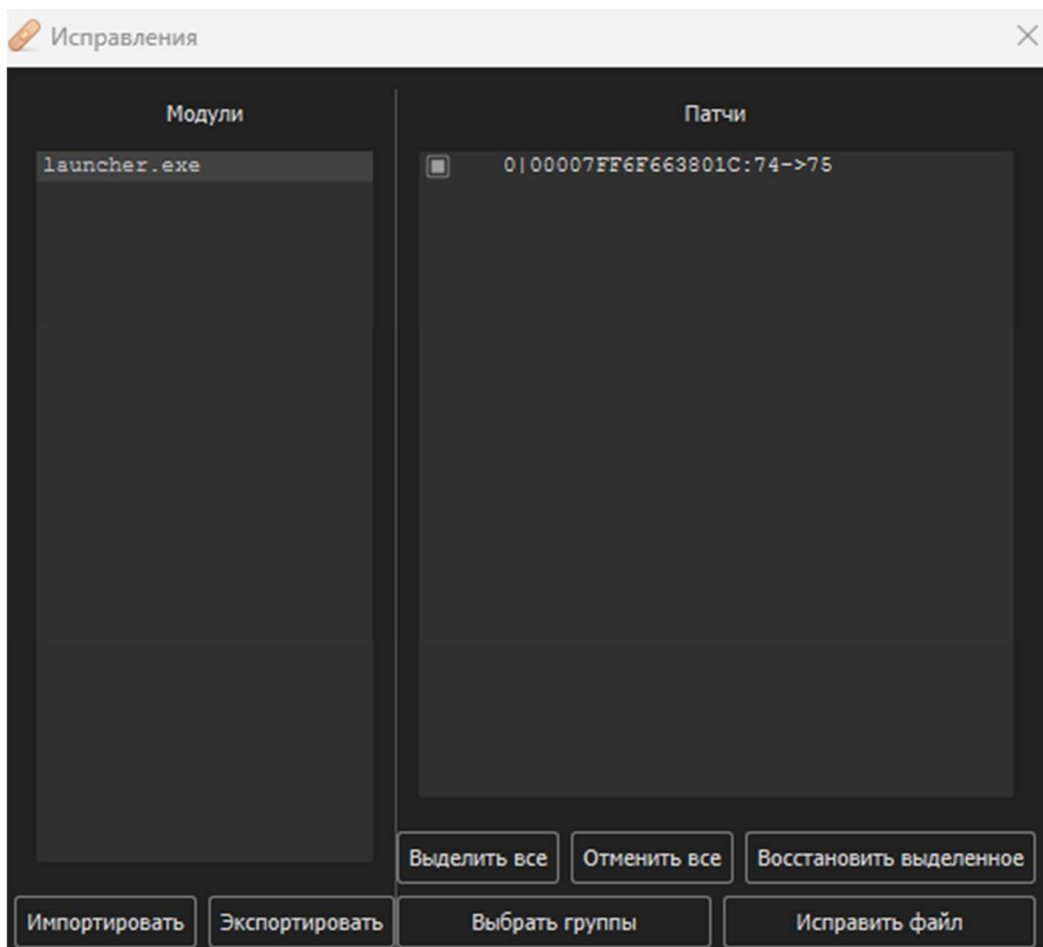
То есть опкод 74 на 75

Жмем Ctrl+E, меняем

Шестнадцатеричное:

```
75 3E
```

Сохраняем исправление



Сохраняем патч как отдельную программу launcher_patched.exe, или заменяем оригинальный лаунчер на него

Запускаем MainProgramm

Вводим любой ключ, получаем флаг

```
Enter key: dasdasd
OK
Access granted. Here is the flag:
FECTF{b4ck_t0_zeroes}
|
```